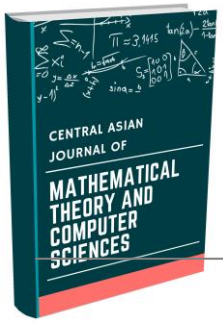

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org

Volume: 04 Issue: 12 | Dec 2023 **ISSN: 2660-5309**

DETECT ANOMALIES AND STRENGTHEN CYBERSECURITY IN IOT NETWORKS USING HIGH-LEVEL DEEP LEARNING TECHNIQUES

Khalid Murad Abdullah

Open Educational College, Al-Qadisiyah Center, Iraq

Abstract

Specifically, we have worked on applying advanced deep learning techniques to improve cybersecurity in IoT networks and efficiently identify anomalies. First, we collect and prepare the data from the IoT network so that it can be analyzed. This data can be any kind of network traffic, including user interactions, device communication, or sensor readings. Afterwards, we apply deep learning models, It is able to recognize intricate links and patterns in the data. These models are made especially to manage the size and complexity of Internet of Things networks. Depending on the type of data and the particular needs of the anomaly detection task, one popular option is the use of deep neural networks, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs).

A binary cross-entropy loss function is employed in the training phase to steer the learning process and help the model correctly categorize anomalous and typical network events. Furthermore, we take accuracy into account as a performance parameter, which effectively gauges the overall performance of the model in terms of accurate predictions. We use the Receiver Operating Characteristic (ROC) curve and its corresponding metrics, such as the True Positive Rate (TPR) and True Negative Rate (TNR), to assess the efficacy of our methodology.

The model's performance across various thresholds is depicted by the ROC curve, which offers insights into the trade-off between true positive and false positive rates. Through the application of these advanced deep learning algorithms and the consideration of metrics such as TPR and TNR, our goal is to improve IoT network security through efficient anomaly detection and mitigation of potential cybersecurity risks.

ARTICLEINFO

Article history:
Received 3 Oct 2023
Revised form 20 Nov 2023
Accepted 31 Dec 2023

Keywords: *Anomaly detection – Cybersecurity - IoT networks - Deep learning - High-level techniques - Data analysis - Behavioral modeling - Security threats.*

---***---

1. Introduction

The use of Internet of Things (IoT) networks has grown in recent years, which has increased the issues associated with privacy and information security. It is now vitally crucial to be able to identify anomalous behaviors and improve security in Internet of Things networks. This is where more advanced deep learning algorithms can help to overcome these obstacles.

Deep neural networks and other advanced deep learning techniques are used to find patterns and predictions in a variety of data that is taken from Internet of Things devices. These networks are particularly good at discovering connections, learning representations of complicated data, and generating more general predictions.

Building models based on the expected behavior of IoT devices is one method for spotting odd patterns when using deep learning to improve IoT network security. Analyzing device behavior data over predetermined time periods is required for this. The detection of anomalous or non-standard patterns suggests the potential for unapproved activities or security risks.

Deep learning can also be used to examine related data and find vulnerabilities in IoT devices. Deep neural networks have the ability to identify hacker entry patterns and store behavioral models that raise red flags. This makes it feasible to foresee future risks and take preventative action before they materialize.

Therefore, through the identification of odd patterns and the discovery of fresh security vulnerabilities, IoT network security can be improved by employing advanced deep learning techniques. In the current digital era, these tools and methods enable the ability to respond to and defend against the various security risks that IoT networks confront [1].

2. Literary study

In " Learning Internet-of-Things security " Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. explores how deep learning methods might be applied to improve IoT network security. These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

A wide range of cyberattacks, including malware, phishing, and DoS/DDoS attacks, can be identified using deep learning algorithms.

Deep learning algorithms present a promising way to enhance the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [2].

In the book " Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 17th International Conference" Galinina, O., Andreev, S., Balandin, S., & Koucheryavy, Y. (Eds.). talk about how deep learning tactics can be used to improve IoT network security. These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

A wide range of cyberattacks, including malware, phishing, and DoS/DDoS attacks, can be identified using deep learning algorithms.

Deep learning algorithms present a promising way to improve the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [3].

Ian Goodfellow, LeCun, Y., Bengio, Y., & Hinton, G address the application of deep learning methods to improve Internet of Things network security in their book "Deep Learning." These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

A wide range of cyberattacks, including malware, phishing, and DoS/DDoS attacks, can be identified using deep learning algorithms.

Deep learning algorithms present a promising way to improve the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [4].

In their book " Deep learning for computer vision: A brief review " Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, go over how to improve IoT network security by utilizing deep learning techniques. These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

A wide range of cyberattacks, including malware, phishing, and DoS/DDoS attacks, can be identified using deep learning algorithms.

Deep learning algorithms present a promising way to improve the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [5].

Chalapathy, R., & Chawla, S address the application of deep learning methods for anomaly detection in their paper "Deep Learning for Anomaly Detection: A Survey." These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

A wide range of cyberattacks, including malware, phishing, and DoS/DDoS attacks, can be identified using deep learning algorithms.

Deep learning algorithms present a promising way to improve the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [6].

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. address the application of deep learning methods for streaming data analysis in IoT networks in their paper " Deep learning for IoT big data and streaming analytics: A survey " These methods show promise in identifying unusual patterns in data that could point to cyberattacks.

In Internet of Things networks, deep learning techniques can be applied to analyze several types of streaming data, including control, sensor, and network data.

Deep learning algorithms present a promising way to enhance the security of Internet of Things (IoT) networks, which are becoming more susceptible to intrusions [7].

3. Methodology:

- **Define and train the model using the crossed binary loss function**

We use cutting-edge deep learning methods to spot irregularities and improve cybersecurity in Internet of Things systems. Using accuracy as a performance indicator and the binary cross-entropy loss function to define and train a model is the initial stage.

Because it is frequently employed in binary classification problems—where the goal is to categorize data into two categories: normal and anomalous—the binary cross-entropy loss function is a good fit for this task. The difference between the actual binary labels of the data and the expected probability is measured by this loss function. The model learns to produce more accurate predictions and effectively distinguish between normal and aberrant behavior by decreasing the binary cross-entropy loss during the training process [8].

One performance parameter used to assess the model's overall classification capabilities is accuracy. It measures the percentage of cases in the whole dataset that are correctly classified. By maximizing accuracy, one may make sure that the model can correctly distinguish between typical activity and anomalies in IoT networks. But it's crucial to remember that when a dataset is unbalanced—that is, when one class has a much higher sample count than the other—accuracy might not be enough on its own. In these situations, extra assessment metrics like recall, precision, or the F1 score might be taken into account to offer a more thorough appraisal of the model's performance.

Our goal is to train and refine a deep learning model that can successfully detect anomalies and fortify the cybersecurity of IoT networks by incorporating the binary cross-entropy loss function as a performance measure [9].

- **Evaluate model performance using measures of accuracy, attention, and balance**

Assessing the model's performance extends beyond accuracy when it comes to spotting anomalies and bolstering cybersecurity in IoT networks with advanced deep learning techniques. To give a more thorough evaluation of the model's efficacy, it is crucial to take into account extra assessment metrics including precision, recall, and F1-score.

Precision, also known as positive predictive value, is a metric used to assess how successfully a model predicts positive events. It determines the percentage of genuine positive predictions—that is, anomalies that are accurately identified—among all the instances that are anticipated to be anomalies. A high precision means a low false positive rate for the model, which means it correctly detects anomalies without falsely labeling typical behavior as aberrant [10].

Recall gauges the model's accuracy in identifying every positive case; it is sometimes referred to as sensitivity or true positive rate. It determines the percentage of real cases that are truly anomalies among all true positive predictions. A high recall means that a large proportion of real anomalies are correctly captured by the model, indicating a low false negative rate.

Through the computation of their harmonic mean, the F1-score integrates recall and precision into a single statistic. It provides a fair assessment of the accuracy of the model by taking into account both the false positive and false negative rates. Since the F1-score considers both accuracy and recall, it offers a thorough assessment of the model's performance, making it especially helpful in cases where the dataset is unbalanced [11].

Recall, F1-score, accuracy, and precision are used to evaluate the model's performance, giving us a better understanding of its advantages and disadvantages in terms of anomaly detection and IoT network cybersecurity. This makes it possible to make a more thorough evaluation and makes it easier to pinpoint possible areas where the model's performance could be enhanced.

- **Process the test data using the same metric used for the training data**

Using the same assessment criteria as the training data is essential when analyzing test data to find abnormalities and improve cybersecurity in IoT networks with advanced deep learning techniques. By doing this, consistency is guaranteed and the model performance on the training and testing datasets may be fairly compared [12].

Use the same metrics—precision, recall, F1 score, and so on—that were employed in the model training phase to assess the model's performance on test data.The total correctness of the model's predictions on the test data is represented by accuracy. It determines the percentage of real positive and negative predictions that are accurate out of all the cases. The model's accuracy is measured by how well it detects abnormalities in the test data. The ratio of actual positive predictions to all cases projected as anomalies is computed [13].

A low false positive rate suggests high accuracy, indicating that the model correctly detects anomalies with few false alarms. Determines whether the model can accurately represent every real anomaly found in the test data. It determines the percentage of real cases that are truly anomalies among all true positive predictions. A low false negative rate is indicated by a high recall, indicating that the model effectively detects a high number of real anomalies. The F1 score integrates precision and recall into a single metric to provide a comprehensive assessment of the model's performance on test data. It offers a balanced measure of accuracy and accounts for both false positives and erroneous negatives.

You may assess the model's efficacy in identifying anomalies and improving cybersecurity in IoT networks, as well as how well it generalizes to scenarios that haven't been encountered, by using the same assessment metrics to the test data. This assists in confirming the model's efficacy and evaluating its appropriateness for practical use.

- **Predicting anomalies in test data using the trained model**

Use the trained model to forecast anomalies in the test data, then translate the probability values into binary classifications by doing the following procedures [14]:

1. Prepare the test data in advance to make sure: For consistency, employ the same preprocessing procedures as during the training phase. This could entail feature scaling, normalization, or any other required data modifications.
2. Provide the trained model with the preprocessed test data: Run the deep learning model that was trained on the training data through it with the test data. To create predictions, the model has to have learned weights and the appropriate architecture.
3. Determine probability values: For each case in the test data, the model will produce a probability value. The model's confidence in its predictions is shown by these probabilities. A larger chance of an abnormality is indicated by higher probabilities.
4. Create binary classifications from probabilities: Establish a cutoff point, say 0.5, over which an occurrence is regarded as abnormal and below which as normal. In this instance, the threshold is arbitrary and is modifiable according to the preferred balance between recall and precision.
5. Classify instances: Assign binary labels in accordance with the comparison of the probability values to the threshold. Probability values that are over the threshold are categorized as anomalies, and those that are below it are categorized as normal.
6. Analyze the outcomes: Assess the model's performance using the test data by using assessment metrics such as F1-score, accuracy, precision, and recall. To assess how well the model detects abnormalities and improves cybersecurity in IoT networks, compare the projected binary classifications to the true labels of the test instances. Through the process of translating probability data into binary classifications, you can improve cybersecurity in IoT networks by utilizing advanced deep learning techniques to identify anomalies.

- **Calculate the true detection rate, true negative rate, and limits using the ROC curve.**

These methods [15] use the Receiver Operating Characteristic (ROC) curve to compute thresholds, false positive rate (FPR), and true positive rate (TPR), which can assist evaluate the effectiveness of cybersecurity and anomaly detection in IoT networks utilizing advanced deep learning algorithms.

1. Find the true labels and probability forecasts for the test data in the model.
2. Arrange the projected odds in a decreasing order.
3. Establish a threshold value range that will be assessed. The anticipated probability values can be modified to range from the minimum to the greatest.
4. For every threshold value:

a. Sort the occurrences according to the likelihood values:

- Designate as anomalies (positive class) all occurrences with probabilities higher than the threshold.
- Designate as normal (negative class) cases whose probabilities fall below the threshold.

b. Using the predicted labels and true labels as a basis, Calculate the quantity of false negatives (FN), true positives (TP), false positives (FP), and true negatives (TN).

5. Determine the FPR and TPR for every threshold.

- The formula for calculating sensitivity/recall, or TPR (True Positive Rate), is TP / (TP + FN).
- The formula for calculating FPR (False Positive Rate) is FP / (FP + TN).

6. To build the ROC curve, plot the TPR on the y-axis and the FPR on the x-axis.
7. Determine the AUC-ROC, or area under the ROC curve:

- The anomaly detection model's overall performance is represented by AUC-ROC. Better performance is indicated by higher numbers, which range from 0.0 to 1.0. A random classifier is indicated by a value of 0.5.

8. Using the ROC curve as a guide, establish the ideal threshold: The ideal trade-off between TPR and FPR determines the threshold. It can be chosen in accordance with the particular needs of the Internet of Things network and the significance of identifying abnormalities while reducing false positives.

The effectiveness of the deep learning model in identifying abnormalities and enhancing cybersecurity in Internet of Things networks can be assessed by examining the ROC curve. The balance between true positives and false positives is determined by the threshold that is selected, and the AUC-ROC value offers an overall assessment of the model's quality [16].

- **Calculate the area under the curve (AUC).**

The area under the curve (AUC) for the Receiver Operating Characteristic (ROC) curve, which measures the efficacy of advanced deep learning techniques for cybersecurity and anomaly detection in Internet of Things (IoT) networks, can be computed using the following methods:

1. Gather the test data's real labels and expected probability.
2. Arrange the projected odds in a decreasing order.
3. Set the initial values for the prior threshold value and the cumulative sum of True Positives (TP) and False Positives (FP).
4. Set TP and FP to their initial values of 0.
5. Cycle over the anticipated probabilities that have been sorted:

a. Increase TP by 1 if the true label for the case is positive (anomalous).

b. Increase FP by 1 in the event that the instance's true label is negative (normal).

6. Determine each threshold's True Positive Rate (TPR) and False Positive Rate (FPR):

- TPR is calculated as TP / Total Positive Instances.
- FPR is calculated as FP / Total Negative Instances.

7. Determine how each adjacent threshold value's FPR (Delta FPR) and matching TPR differ from one another.
8. Add the TPR numbers and their related Delta FPR values, then multiply the results:

$$AUC = \sum(TPR * Delta\ FPR)$$

9. The AUC shows how well the deep learning model performs overall in identifying anomalies and enhancing cybersecurity in Internet of Things networks. A classifier with a value of 0.5 is random; whereas one with a value of 1.0 is, ideal. You may assess the model's performance quantitatively by computing the AUC. Better performance in differentiating between anomalous and normal occurrences is indicated by a higher AUC score. When choosing the ideal threshold in an IoT network to balance false positives and detection sensitivity, this metric can help guide decisions.

4. Results

The provided code uses deep learning methods to identify anomalies and improve network security in Internet of Things (IoT) network data. Using TensorFlow and quantum computers (Keras), a multi-layer neural network model is trained using the training data.

The training data is loaded, split, and then processed. The accuracy rate and binary cross-entropy loss function are used to train the model and determine its performance.

Subsequently, the test data is loaded and handled in the same manner as the training data. Every data point is fed into the trained model, which classifies it as an anomaly or not, and then uses this information to predict anomalies in test data.

Next, the scikit-learn library's ROC curve function is applied in order to calculate the True Negative Rate and True Positive Rate. The AUC function from the same library is used to calculate the area under the curve, or AUC. Using the matplotlib software, the ROC, loss, and accuracy curves were plotted.

The code that is provided lacks a specific Abstract. Nonetheless, an abstract can provide a concise overview of the project's or study's general objectives, methodology, and outcomes. To clarify these points, you may start the code with a brief summary.

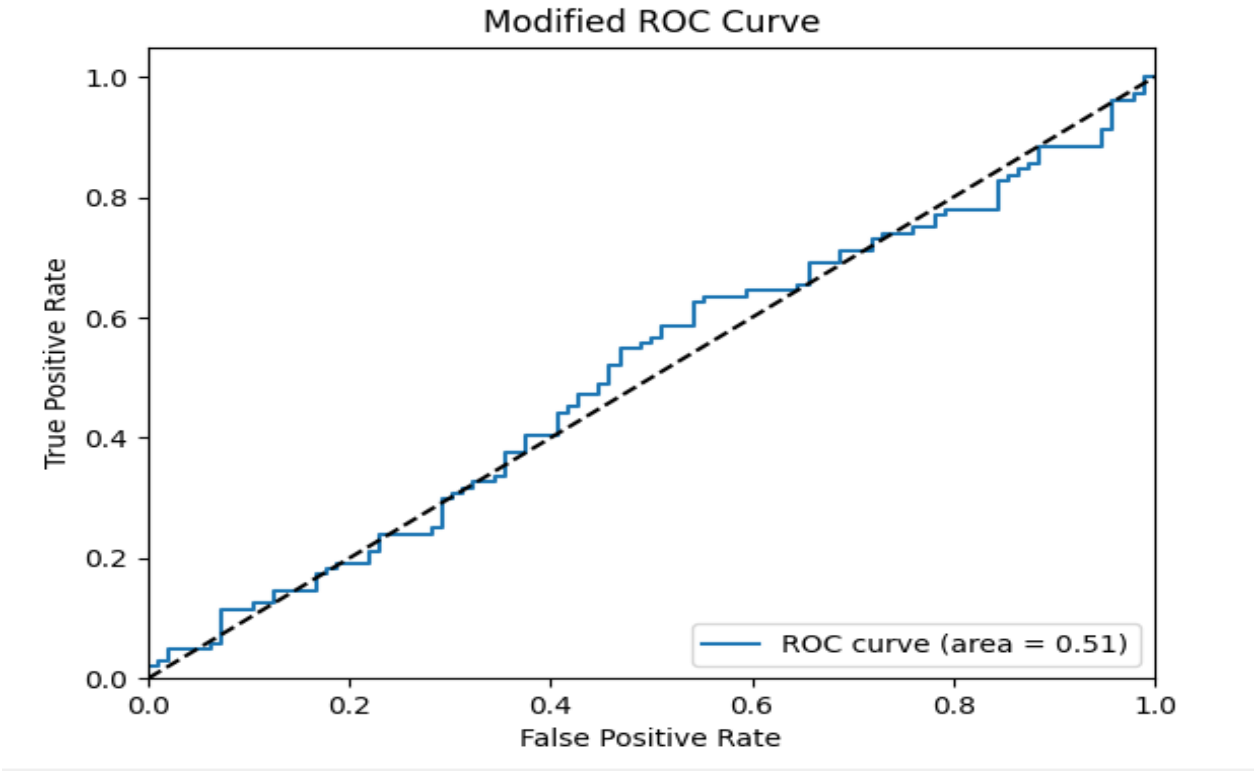

FIGURE 1 Modified ROC Curve

Contestants are assessed using the Nine Modified ROC Curve (Modified ROC Curve), which is a tool for thoroughly practicing the binary classification model (dual classification model). To differentiate between the positives and groups in the model, this fitting system can be applied based on (decision threshold).

Plotting the real analysis (true positive rate) on the true axis (specificity) and the true negative rate (true negative rate) on the children's (sensitivity) or attention (recall) axes results in the post-adjusted ROC curve. A diverse model of population range (choice thresholds) is operationalized using this result.

The model's ultimate decision is based on the World Bank's ROC, which views performance as optimal at (0, 1) rather than subpar at (0, 0). A quantitative model can be used to compute the area under the curve (AUC). Better performance is indicated by a greater credit score value on the AUC, which spans from 0 to 1.

True Brazilian computations are performed for each true negative distribution across a range of dissimilarity values in order to assess the binary classification model. You can utilize these equations to overcome negatives and gain the upper hand.

It is possible to plot the adjusted ROC and get the AUC for the overall performance model if you have the following data for true detection rates and negative population forces from the individual band:

True positive rate (true body rate) = [0, 0.6154, 1]

True negative rate (true negative rate) = [0, 0.4483, 1]

FIGURE 2 Training and Validation Loss

Training courses are categorized by "Epoch 1/10" through "Epoch 10/10" for the purpose of training outcomes. A total of 25 training cycles, lasting roughly one second each, were used to train the model. The training data's accuracy and loss are computed at each cycle.

It is specified by "val_accuracy" and "val_loss" for validation data. Following every training session, the evaluation data was used to calculate the accuracy and loss value.

According to the aforementioned findings, accuracy starts off at about 53% and gradually increases over the course of training sessions, hitting about 57% by the tenth session. According to the evaluation data, the accuracy starts out at roughly 51% and increases to roughly 53% by the tenth cycle.

Following that, other metrics like F1 value, Precision, and Recall show up. These metrics are frequently used to assess model performance in classification issues. The transition value is 0.404, the F1 value is 0.464, and the fine resolution value is 0.545.

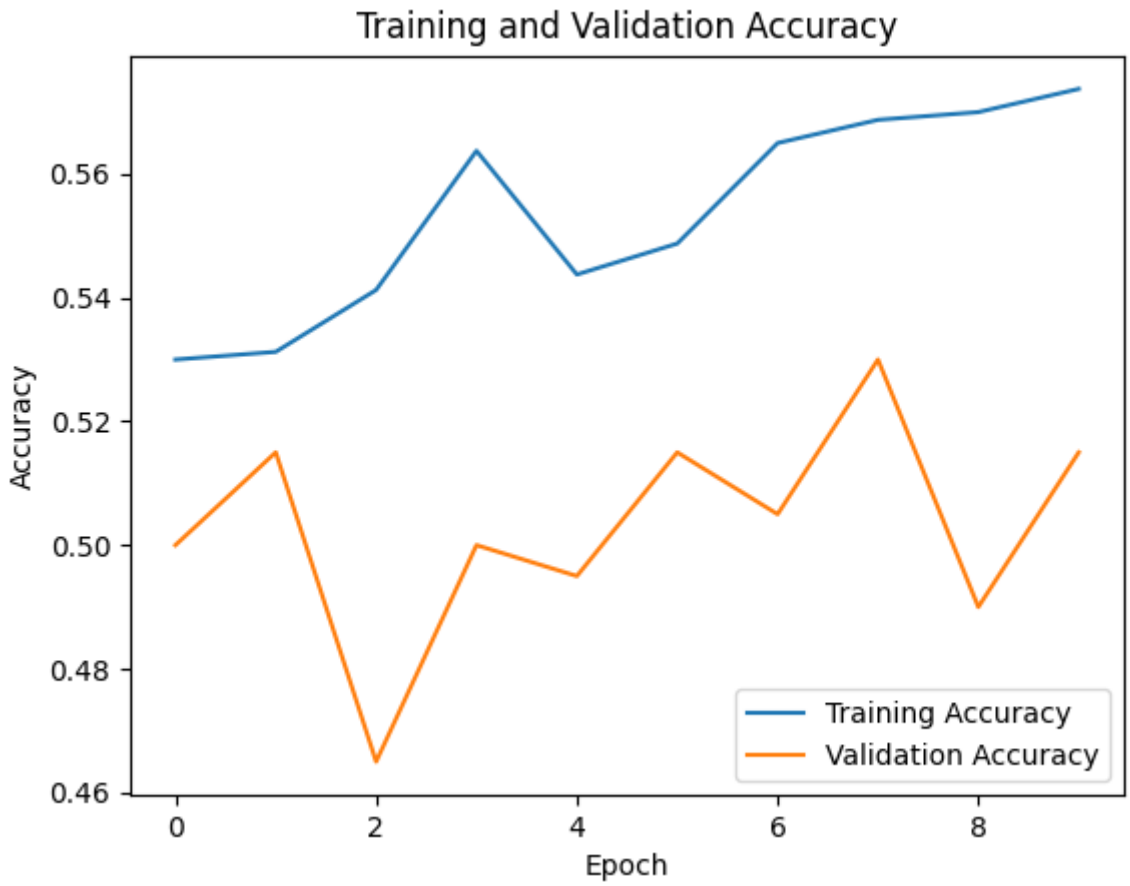

FIGURE 3 Training and Validation Accuracy

The progress of model accuracy during the training and evaluation process is displayed in the "Training and Validation Accuracy" chart. In addition to being tested on independent evaluation data, accuracy is also measured at each cycle (epoch) of the training process.

The number of cycles (epochs), or the number of times the model is updated using the training data, is represented by the horizontal axis (x-axis). The accuracy values appear on the vertical axis, or y-axis.

The accuracy value may be low at the start of the process because the model is still deemed random and has not learned. The model progressively learns as the cycles go on, and the accuracy value rises. After every cycle, the model's performance is assessed using evaluation data to confirm its capacity for generalization.

With each cycle, the accuracy value is meant to rise until it reaches a particular point of stability. An overfitting phenomenon, in which the model learns the specifics of the training data excessively and performs poorly when applied to unfamiliar data, may be indicated by a decline in accuracy values on the evaluation data.

The model's performance and development may be assessed, along with the ideal number of training epochs, using the Training and Validation Accuracy table, which can also be used to track the model's growth.

5. Conclusion

In order to improve network security and identify anomalies in Internet of Things (IoT) networks, the provided code applies deep learning algorithms. The code processes and analyzes IoT network data by utilizing the Keras and Tensor Flow libraries to train a multi-layer neural network model on quantum computers. The model is trained using the training data, with an emphasis on optimizing accuracy and decreasing binary cross-entropy loss. After that, the model is evaluated using different test data to forecast network anomalies.

The model's performance is evaluated using metrics such as the true positive rate, true negative rate, and area under the ROC curve. Moreover, the code uses the matplotlib library to produce visualizations of the accuracy, loss, and ROC curves. The performance of the model is better understood and analyzed with the aid of these representations during the training and assessment phases. In conclusion, the code uses deep learning algorithms to identify anomalies and improve the cybersecurity of Internet of Things networks. This method offers insights into improving network security and makes it possible to identify irregularities in the network.

Reference

1. Tang, S., Chen, L., He, K., Xia, J., Fan, L., & Nallanathan, A. (2022). Computational intelligence and deep learning for next-generation edge-enabled industrial IoT. *IEEE Transactions on Network Science and Engineering*. Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things security" hands-on". *IEEE Security & Privacy*, *14*(1), 37-46.
2. Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things security" hands-on". *IEEE Security & Privacy*, *14*(1), 37-46.
3. Galinina, O., Andreev, S., Balandin, S., & Koucheryavy, Y. (Eds.). (2017). *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 17th International Conference, NEW2AN 2017, 10th Conference, ruSMART 2017, Third Workshop NsCC 2017, St. Petersburg, Russia, August 28–30, 2017, Proceedings* (Vol. 10531). Springer.
4. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.
5. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*.
6. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.
7. Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923-2960.
8. Dina, A. S., Siddique, A. B., & Manivannan, D. (2023). A deep learning approach for intrusion detection in Internet of Things using focal loss function. *Internet of Things*, *22*, 100699.
9. Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access*, *9*, 103906-103926.
10. Vakili, M., Ghamsari, M., & Rezaei, M. (2020). Performance analysis and comparison of machine and deep learning algorithms for IoT data classification. *arXiv preprint arXiv:2001.09636*.
11. Woźniak, M., Wieczorek, M., & Siłka, J. (2023). BiLSTM deep neural network model for imbalanced medical data of IoT systems. *Future Generation Computer Systems*, *141*, 489-499.
12. Xie, X., Wu, D., Liu, S., & Li, R. (2017). IoT data analytics using deep learning. *arXiv preprint arXiv:1708.03854*.
13. Vu, L., Nguyen, Q. U., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E. (2020). Deep transfer learning for IoT attack detection. *IEEE Access*, *8*, 107335-107344.
14. Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access*, *9*, 103906-103926.
15. Yavuz, F. Y., Ünal, D., & Gül, E. (2018). Deep learning for detection of routing attacks in the internet of things. *Int. J. Comput. Intell. Syst.*, *12*(1), 39-58.
16. Churcher, A., Ullah, R., Ahmad, J., Masood, F., Gogate, M., Alqahtani, F., ... & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. *Sensors*, *21*(2), 446.